

Tartu Ülikool

Matemaatika- Informaatika Teaduskond

Arvutiteaduse Instituut

Joosep Kibal

**Tekstülesannete lahendamise programmi
funktsionaalsuse täiendamine**

Bakalaureusetöö

Juhendaja: dotsent Rein Prank

Tartu

2013

Sisukord

Sissejuhatus	3
1. Programmi varasema versiooni kirjeldus ja puudused	6
1.1. Ülesande lahendamine	6
1.1.1. Küsimuse moodustamine	6
1.1.2. Avaldise moodustamine	7
1.1.3. Arvutamine	9
1.2. Olemasoleva rakenduse kitsendused	11
Bakalaureusetöö käigus tehtud muudatused	12
2. Lahendamise programmi tehtud muudatused	13
2.1. Ülesande lahendamine	13
2.1.1. Küsimuse moodustamine kahe paneeliga	13
2.1.2. Avaldise moodustamine kasutades jagamistehet	14
2.1.3. Kümnnendmurdudega arvutamine	14
2.2. Lahenduskäigu salvestamine	15
2.3. Vigade salvestamine	16
2.4. Tekstülesannete koostamise liides	18
2.4.1. Ülesande teksti koostamine ja muutujate lisamine	18
2.4.2. Ülesande küsimuste moodustamine	19
2.4.3. Küsimustele vastavate avaldiste ja vastuse moodustamine	21
2.4.3 Moodustatud seoste kuvamine	22
2.4.4. Ülesande salvestamine	22
2.4.5. Ülesande avamine	23
2.4.6. Uue ülesande loomine	23
2.4.7 Muutujate genereerimine	23
3. Ülesande esitus tekstifailis	24
Programmi ja andmestruktuuride lühike kirjeldus	26
4. Programmi edasi arendamise võimalused	27
Adding functionality to word problem solving environment	28
Viited	29
Lisa 1	30

Sissejuhatus

Käesoleva bakalaureusetöö sisuks on tekstülesannete lahendamise programmi TEKSTER funktsionaalsuse laiendamine. TEKSTERi baasversiooni [1] on loonud Evari Koppel 2006. aastal bakalaureusetöö raames.

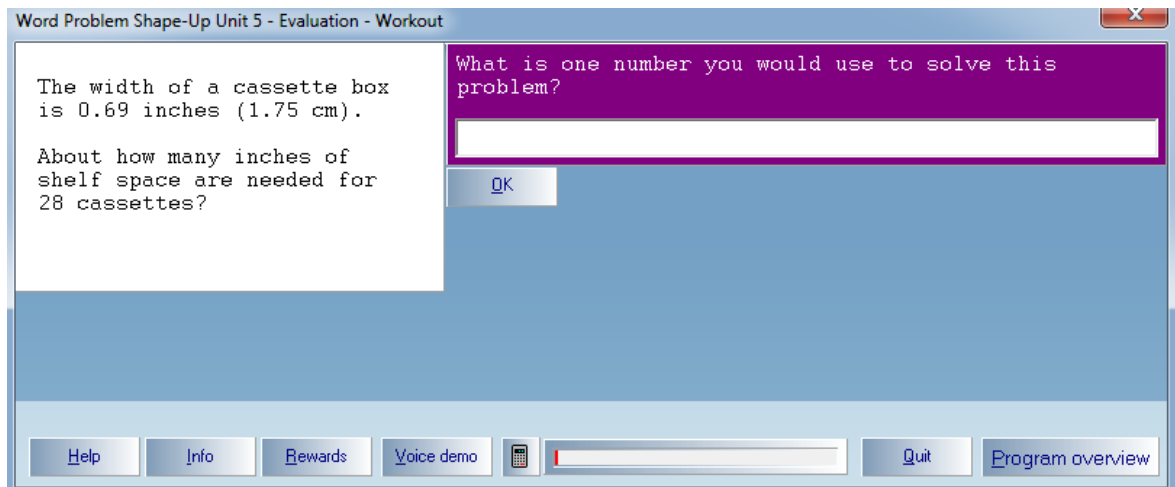
Töö käigus kirjeldatakse TEKSTERi baasversiooni lisatud täiendusi. Samuti seletatakse nii käesoleva töö käigus valminud ülesannete koostamise programmi ülesehitust kui ka andmete esitust.

Töö koosneb neljast peatükist. Esimeses peatükis kirjeldatakse baasversiooni kitsendusi ja tuuakse välja rakendusele tehtud muudatused. Teises peatükis tutvustatakse ülesannete koostamise programmi kasutajaliidest ja TEKSTERisse tehtud muudatusi. Kolmandas peatükis kirjeldatakse programmi andmestruktuure ja ülesandefaili. Neljandas peatükis tutvustatakse, millised on võimalused rakendust edasi arendada.

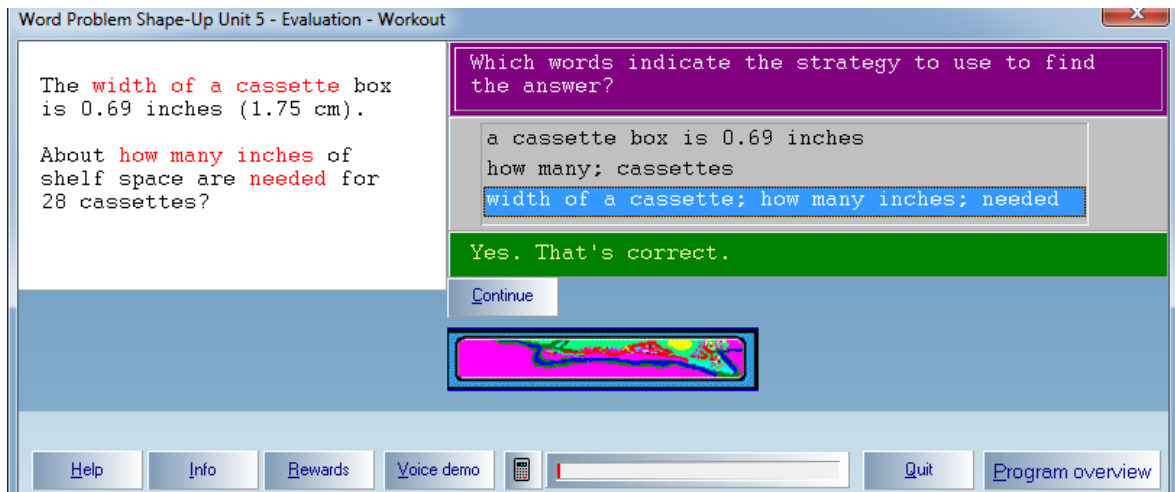
Lisana esitatakse täiendatud programm, ülesannete koostamise programm ja uuendatud ülesannete kogumik [2].

Lahendamise programmi uue variandi programmeerimise eel võrreldi loodud programmi teiste samalaadsete programmidega. Leidus üks võrreldav toode, millel on sarnased võimalused – *Merit software Word Problem Shape-Up* [3]. „Word Problem Shape-Up“ võimaldab lahendada tekstülesandeid kasutades liitmist, lahutamist, korrutamist ja jagamist. Sarnaselt TEKSTERile võimaldab „Word Problem Shape-Up“ samm haaval lahendamist. Erinevalt tekstierist küsib „Word Problem Shape-Up“ kasutajalt, milliseid väärtusi ülesande lahendamiseks vaja on (joonis 1) Lahendajalt küsitakse, millist ülesande tekstis olevat arvu tuleks ülesande lahendamiseks kasutada (küsitakse nii mitut arvu kui lahendamiseks vaja on – kui ülesande saab lahendada kahe arvuga küsitakse kahte arvu jne). Programm pakub kasutajale valiku milliste küsimuste abil ülesande vastuseni jõuda (joonis 2). Programm laseb kasutajal valida, mis tehte või tehete ülesannet tuleks lahendada (joonis 3) ja seletab kasutajale kuidas ülesannet lahendada. Lisaks pakub „Word Problem Shape-Up“ kasutajale ette vastusevariante ja näitab lahenduskäiku. TEKSTERis peab kasutaja moodustama ise küsimused ja leidma neile vastavad avaldised ja vastused. Lisaks on TEKSTERile võimalik kasutajal ülesandeid juurde teha, aga „Word Problem Shape-Up“ koosneb kolmest moodulist, milles igas on 180 tekstülesannet. „Word Problem Shape-Up“

kahjuks räägib ka see, et ta on tasuline ja prooviversioonis saab katsetada ainult mõningaid võimalusi.



Joonis 1. Ülesande tekstist lahendamiseks vajaliku arvu sisestamine



Joonis 2. Lahendamiseks sobivate küsimuste valimine

Word Problem Shape-Up Unit 5 - Evaluation - Workout

A 14-ounce jar of honey was 6 inches (15.2 cm) high. It contained 8.4 ounces of clover honey, 3.7 ounces of honeysuckle honey, and the rest marked "from other sources." (1 ounce equals 29.6 milliliters.)


About how much honey (in ounces) was "from other sources"?


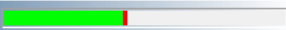
What operation is best used to solve this problem?

Subtraction/Multiplication
 Multiplication/Division
 Not given
Addition/Subtraction

Yes. The amount from clover and honeysuckle is found by ADDING. The DIFFERENCE between that sum and the contents of the jar is how much honey comes from other sources.

Continue



Help Info Rewards Voice demo   Quit Program overview

Joonis 3. Lahendamiseks sobiva või sobivate operantide valimine

1. Programmi varasema versiooni kirjeldus ja puudused

Algklassides toimub esimest tüüpi tekstülesannete lahendamine samm sammu haaval. Iga tehte jaoks tuleb selgeks teha, mida antud samm tähendab. Selleks koostatakse tehte kohta küsimus või selgitus. Algul leitakse koostatud küsimusele vastus ühe tehte abil, teadmiste süvenedes aga juba mitmega. Näiteks:

Ülesanne:

Toomas on 3 aastane, tema vend Taavi on 4 aastat vanem.

Nende ema on 3 korda vanem kui Toomas ja Taavi kokku.

Kui vana on ema?

Lahendus:

1.Kui vana on Taavi? $3+4=7$

2.Kui vanad on Toomas ja Taavi kokku? $3+7=10$

3.Kui vana on ema? $10*3=30$

Järgnevalt refereerime E. Koppeli bakalaureusetöö [1] järgi kuidas toimib tekstülesannete lahendamine TEKSTERi baasvariandis.

1.1. Ülesande lahendamine.

Ülesande lahendamise iga samm koosneb kolmest etapist: küsimuse moodustamine, avaldise moodustamine ning arvutamine.

1.1.1. Küsimuse moodustamine.

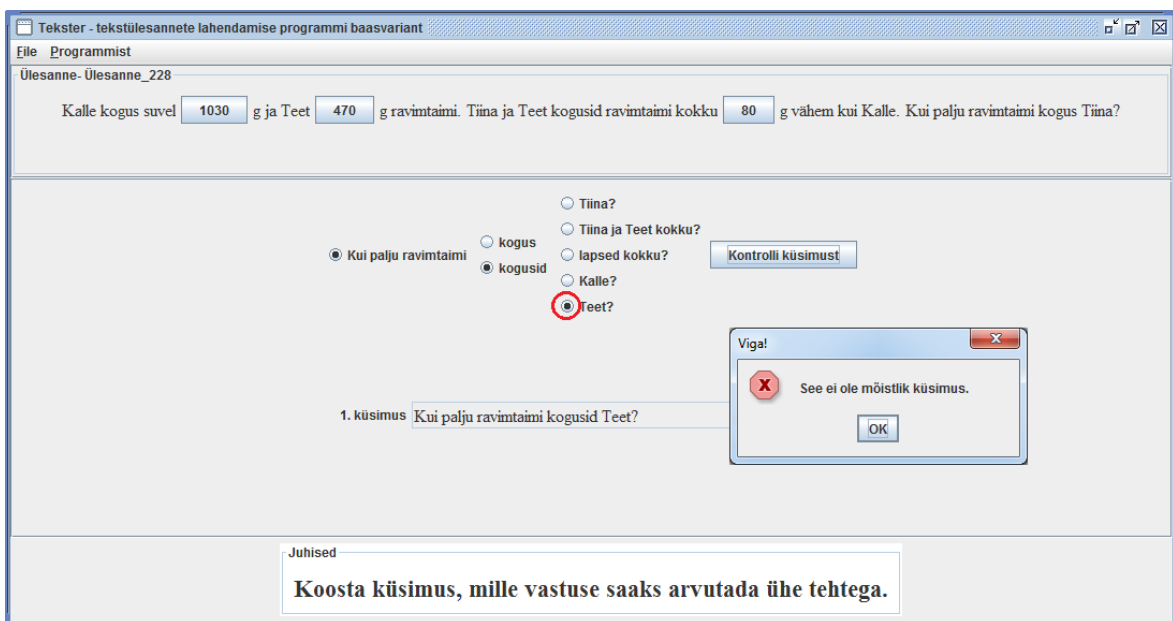
Kasutaja moodustab talle ette antud lause osade maatriksi abil küsimuse, millele ta selle sammuga vastust soovib leida. Seejärel vajutab nupule „Kontrolli küsimust” (joon. 5).

Joonis 5.

Programm kontrollib selle etapi juures, kas:

- a. Moodustatud küsimus ei ole mõistlik – see tähendab, et õpetaja pole sellele küsimusele seadnud vastavusse mingit avaldist;
- b. Sellele küsimusele on juba vastus olemas – vastus on algandmete hulgas või saadud eelmistel sammudel;
- c. Sellele küsimusele ei saa ühe tehtega vastust leida.

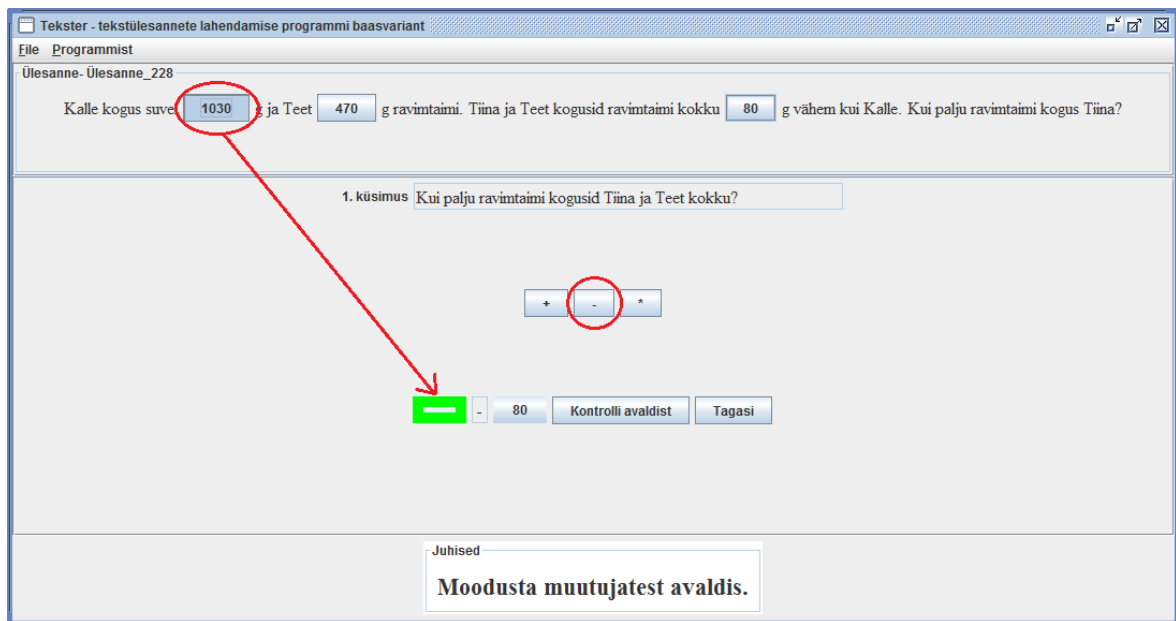
Juhul, kui kontrollimise käigus selgub, et tegu on mõne ülal mainitud juhuga, annab programm sellest teada veateatega (joon. 6). Kui aga moodustatakse küsimus, millele saab ühe tehtega leida vastuse ning millele on õpetaja seadnud vastavusse ülesandefailis mingi avaldise, siis liigutakse edasi järgmise etapi juurde.



Joonis 6.

1.1.2. Avaldise moodustamine.

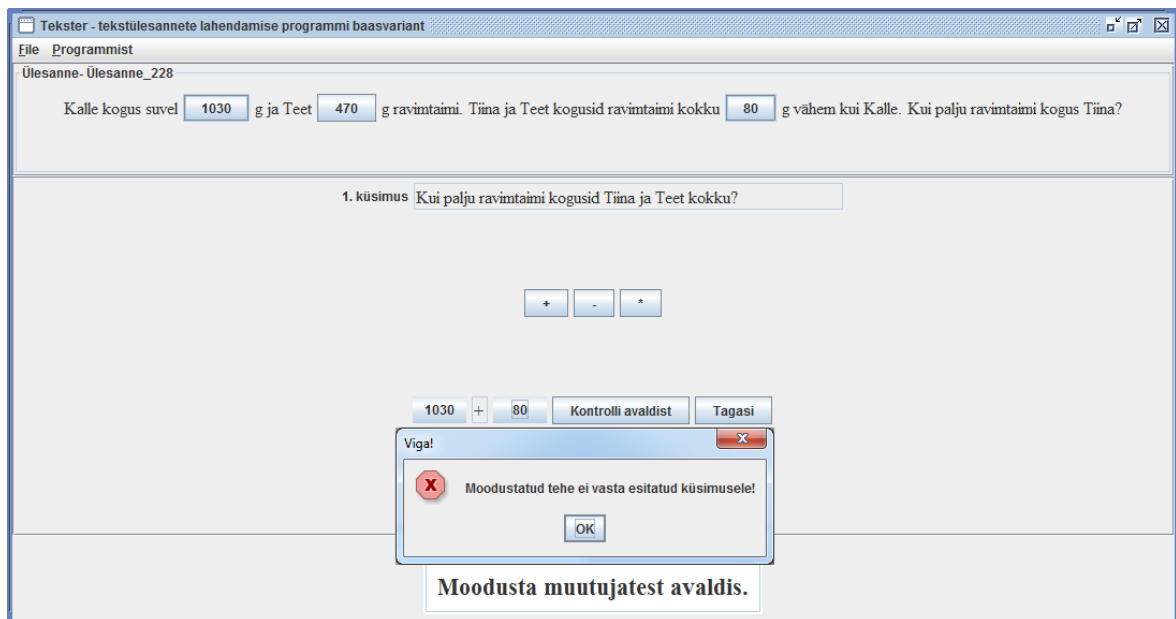
Kasutaja moodustab ülesandes antud ja eelmistel sammudel leitud suurustest küsimusele vastava avaldise. Tekkinud kastidesse tuleb arvud „lohistada”, vajutades selleks suurust kujutava nupu kohal hiire vasak nupp alla, seda all hoides lohistada kursor sihtnupu kohale ning seejärel nupp lahti lasta. Programm annab teada sobivast sihtpunktist rohelise kastiga. Lisaks muutujaile tuleb määrata ka tehtemärk. Selleks tuleb teha vastava märgi peal hiirega klikk (joonis 7).



Joonis 7.

Selle etapi juures kontrollitakse:

- d. Tehtemärgi olemasolu avaldises – märgi puudumisel teavitatakse veateatega;
- e. Arvude olemasolu avaldises – muutujate puudumisel teavitatakse veateatega;
- f. Avaldise vastavust küsimusele – kui moodustatud tehe ei vasta küsimusele, siis teavitatakse sellest kasutajat veateatega (joon. 8).

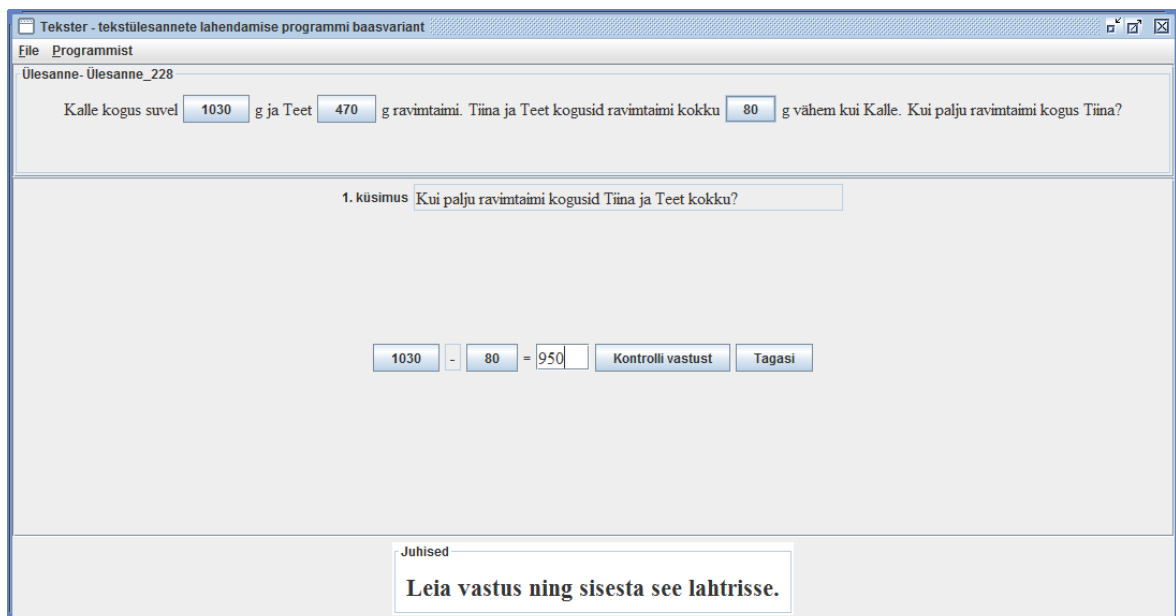


Joonis 8.

Kui avaldis vastab moodustatud küsimusele, siis liigutakse edasi kolmanda etapi juurde – arvutamine.

1.1.3. Arvutamine.

Kasutaja peab arvutama avaldise väärtuse. Väärtuse lahtrisse saab sisestada vaid arve (joon. 9). Kui kasutaja eksib arvutamisel, siis programm annab sellest teada veateatega.



Joonis 9.

Küsimuse õige vastuseni jõudmise järel küsib programm, kas kasutaja soovib ülesannet edasi lahendada või lõppvastust anda (joon. 10). Kui kasutaja eksib

(näiteks soovib edasi lahendada, kuigi on juba lõppvastuse leidnud), siis programm teavitab sellest kasutajat ning laseb tal uuesti otsustada.

Tekster - tekstülesannete lahendamise programmi baasvariant

File Programmist

Ülesanne- Ülesanne_228

Kalle kogus suvel 1030 g ja Teet 470 g ravintaimi. Tiina ja Teet kogusid ravintaimi kokku 80 g vähem kui Kalle. Kui palju ravintaimi kogus Tiina?

1. küsimus Kui palju ravintaimi kogusid Tiina ja Teet kokku?

1030 - 80 = 950

2. küsimus

Vastuse andmine

Kas sa soovid ülesannet edasi lahendada?

Jah, soovin edasi lahendada. Ei, see oli lõppvastus.

950 - 470 = 480 Kontrolli vastust Tagasi

Juhised

Leia vastus ning sisesta see lahtrisse.

Joonis 10.

Ülesande esitus tekstifailis

Iga ülesanne esitatakse spetsiaalse kujundusega tekstifailis. Faili laiendiks peab olema .txt.

Failis esitatakse:

1. Ülesande tekst, kus lahenduses kasutatavad suurused märgitakse muutujatena;
2. Küsimuste moodustamise maatriksi veerud;
3. Muutujatele ja neist moodustatud avaldistele vastavad küsimused;
4. Vastuse avaldis.

Näiteks:

-=TEKSTER=-

!TEKST=

Teises klassis õpib #a=8 poissi ja #b=9 tüdrukut.

Kolmandas klassis õpib #c=7 poissi ja #d=10 tüdrukut.

Mitu õpilast õpib teises ja kolmandas klassis kokku?

```

!VEERUD=
Mitu
poissi&tüdrukut&õpilast
õpib&magab
teises klassis?&kolmandas klassis?&teises ja kolmandas klassis?
!VASTAVUSED=
a=Mitu poissi õpib teises klassis?
b=Mitu tüdrukut õpib teises klassis?
c=Mitu poissi õpib kolmandas klassis?
d=Mitu tüdrukut õpib kolmandas klassis?
a+c=Mitu poissi õpib teises ja kolmandas klassis?
b+d=Mitu tüdrukut õpib teises ja kolmandas klassis?
a+b=Mitu õpilast õpib teises klassis?
c+d=Mitu õpilast õpib kolmandas klassis?
(a+b) + c+d=Mitu õpilast õpib teises ja kolmandas klassis?
!VASTUS=
a+b+c+d

```

1. Ülesande faili tuvastus – ülesande fail peab algama reaga -=TEKSTER=-.
2. Ülesande tekst ning arvud selles – programm kuvab kasutajale ülesande teksti ning selles esinevad muutujad on välja toodud nuppudena. Õpetaja või ülesannete sisestaja peab ülesandefailis kõik vajalikud muutujad vastavalt nõuetele tähistama.

Õpilasele kuvatakse ülesande tekst ilma muutujateta. Muutujate väärtused on ekraanil nuppudena.

3. Küsimuste moodustamise maatriks – maatriksi abil saab õpilane moodustada küsimusi ülesande lahendamiseks.
4. Küsimustele vastavad avaldised – programm vajab lahenduskäigu kontrollimiseks ülesande algandmeid tähistavatest muutujatest koostatud avaldiste ning küsimuste vahelisi vastavusi.
5. Vastuse avaldis – programm vajab vastuse avaldist, et kindlaks teha, millal on ülesanne lahendatud.

Ülesande avamisel kontrollib programm eelpool toodud tingimusi ja nende mitte täitmise korral annab sellest kasutajale teada vastava sisulise veateatega ning ei ava ülesannet.

1.2 Olemasoleva rakenduse kitsendused

Olemasoleva rakenduse kitsendused on:

- Saab lahendada ainult positiivsete täisarvudega liitmist, lahutamist ja korrutamist sisaldavaid ülesandeid.

- Puudub ülesande koostamise liides. Kõik ülesanded tuleb koostada mingit üldotstarbelist tekstiredaktorit kasutades jälgides sealjuures etteantud süntaksit.
- Rakendus ei võimalda salvestada ülesande lahendaja nime, lahenduskäiku ja lahendaja tehtud vigu.

Bakalaureusetöö käigus tehtud muudatused

Käesoleva bakalaureusetöö käigus lisati rakendusele järgnevad võimalused:

- Jagamistehe – võimaldab lahendada tekstülesandeid, mis sisaldavad jagamistehteid.
- kümnendmurdudega arvutamine – võimaldab lahendada tekstülesandeid, mis sisaldavad kümnendmurde.
- lahenduskäigu salvestamine – programm salvestab kasutaja lahendus käigus tehtud sammud lahendusfaili.
- lahendaja vigade salvestamine – programm salvestab kasutaja tehtud vead lahendusfaili ja eraldi vigadefaili.
- küsimuste moodustamine viidi kahte paneeli – programmis on võimalik küsimusi moodustada kahe paneeli abil.
- ülesannete koostamise liides – õpetajaliides tekstülesannete koostamiseks.
- ülesannete koostamise liidesesse lisati lähteandmete genereerimise võimalus – võimaldab lahendajaprogrammis genereerida igakordsel avamisel uued lähteandmed.

2. Lahendamise programmi tehtud muudatused

2.1. Ülesande lahendamine

Ülesande lahendamise samm koosneb kolmest osast: küsimuse moodustamine, avaldise moodustamine ja arvutamine.

2.1.1. Küsimuse moodustamine kahe paneeliga.

Kasutaja moodustab ette antud lause osade maatriksi abil küsimuse, millele ta vastust soovib leida. Esimesena kasutaja valib millise maatriksi abil soovib ta küsimust moodustada (joonis 4). Selleks vajutab ta vastava maatriksi ees olevasse märkeruutu (vaikimisi on valitud ülemise maatriksi ees olev ruut). Seejärel vajutab kasutaja nupule „Kontrolli küsimust“

Joonis 4. Küsimuste moodustamine mitme paneeli abil

Programm võimaldab selle etapi juures moodustada küsimuse korraga ainult ühe maatriksi kaudu. Kasutaja võib küsimuse moodustamise ajal vabalt vahetada, millise maatriksi abil ta küsimust moodustab. Kui kasutaja vahetab maatriksit (vajutab hiirega „Vali“ märkeruudule), siis programm muudab valitud maatriksi aktiivseks ja teise maatriksi mitteaktiivseks ja eemaldab kasutaja poolt eelnevalt valitud küsimuse osad.

2.1.2. Avaldise moodustamine kasutades jagamistehet.

Kasutaja valib tehtemärkide hulgast jagamise. Selleks tuleb jagamismärgi peal teha hiirega klikk (joonis 5).

The screenshot shows a window titled "Tekster - tekstülesannete lahendamise programmi baasvariant". Inside, there's a text area with the following content:

Teises klassis õpib 5 poissi ja 9 tüdrukut. Kolmandas klassis õpib 9 poissi ja 10 tüdrukut. Kokku on lõunaks 200 mandariini. Iga õpilane sai võrdse koguse mandariine. Mitu mandariini sai iga õpilane?

Below the text are four questions:

1. küsimus Mitu poissi õpib teises ja kolmandas klassis?
5 + 9 = 14
2. küsimus Mitu tüdrukut õpib teises ja kolmandas klassis?
9 + 10 = 19
3. küsimus Mitu õpilast õpib teises ja kolmandas klassis?
14 + 19 = 33
4. küsimus Mitu mandariini sai iga õpilane?
200 : 33

The division symbol (:) in the fourth question is circled in red. At the bottom, there are buttons for "Kontrolli avaldist" and "Tagasi". A status bar at the very bottom says "Juhised" and "Moodusta muutujatest avaldis."

Joonis 5 avaldise moodustamine kasutades jagamistehet

Selle etapi juures täiendati avaldise vastavuse kontrolli. Programm teavitab kasutajat veateatega ,kui moodustatud jagamistehe ei vasta küsimusele.

2.1.3. Kümnnendmurdudega arvutamine.

Kasutaja peab arvutama avaldise väärtuse. Kui kasutaja eksib arvutamisel, siis väljastab programm veateate. Kasutaja saab sisestada ainult numbreid ja kümnnendpunkti sümbolit. Vastus peab olema arvulisel kujul numbritena sisestatud ja komakoha eraldajaks on kümnnendpunkt. Kasutaja peab vastuse andma kümnnendiku täpsusega, sest programm ümardab kontrollimise käigus vastuse murdosa kümnnendiku täpsusega. (joonis 7).

Tekster - tekstülesannete lahendamise programmi baasvariant

File Programmist

Ülesanne- jagamine_murd

Teises klassis õpib 5 poissi ja 9 tüdrukut. Kolmandas klassis õpib 9 poissi ja 10 tüdrukut. Kokku on lõunaks 200 mandariini. Iga õpilane sai võrdse koguse mandariine. Mitu mandariini sai iga õpilane?

1. küsimus Mitu poissi õpib teises ja kolmandas klassis?

5 + 9 = 14

2. küsimus Mitu tüdrukut õpib teises ja kolmandas klassis?

9 + 10 = 19

3. küsimus Mitu õpilast õpib teises ja kolmandas klassis?

14 + 19 = 33

4. küsimus Mitu mandariini sai iga õpilane?

200 : 33 = 6.1

Kontrolli vastust Tagasi

Juhised

Leia vastus ning sisesta see lahtrisse.

Joonis 6 vastuse arvutamine kümnendiku täpsusega

Õige vastuse korral küsib programm, kas kasutaja soovib ülesannet edasi lahendada või lõppvastust anda. Nulliga jagamisel väljastatakse kasutajale veateade (joonis 7). Nulliga jagamine võib tekkida, kui koostamise käigus on muutuja väärtuseks pandud null või lahendamise käigus tekib mingi tehte tulemusena avaldise väärtuseks null (selle võimaluse esinemistõenäosus on suurem kui muutujad genereeritakse juhuslikult).

Tekster - tekstülesannete lahendamise programmi baasvariant

File Programmist

Ülesanne- nulljagamine

Auto sõidab 105 km tunnis. Kui pika tee sõidab auto 0 tunniga?

1. küsimus Kui pika tee sõidab auto?

Viga!

Jagamine nulliga! Ülesanne on vigane.

OK

105 : 0

Kontrolli avaldist Tagasi

Juhised

Moodusta muutujatest avaldis.

Joonis 7 avaldise kontroll nulliga jagamisel

2.2. Lahenduskäigu salvestamine

Iga ülesande kohta luuakse lahendusefail, kuhu programm kirjutab kõik ülesande lahendamiseks tehtud sammud ja nende tulemused.

Lahendusefail kirjutatud üks lahenduskäik on kujul:

1. Uue ülesande algust tähistav rida „=UUS“ <lahendaja nimi, ülesandefaili avamise kuupäev ja kellaaeg>.
2. Ülesande nimi
3. Kasutaja lahenduskäik
4. Lahenduste lõppu tähistav rida „Suurepärase! Ülesanne lahendatud. . <vastuse andmise kuupäev ja kellaaeg>“

Programmi avades küsitakse kasutajalt tema nime, mis kirjutatakse lahendusefaili „=UUS“ järele. Kasutaja ei pea sisestama nime, sellisel juhul jäetakse lahendaja nime osa tühjaks. Programm kirjutab lahendusefaili rea „Suurepärase! Ülesanne lahendatud. <vastuse andmise kuupäev ja kellaaeg>“ ainult siis ,kui kasutaja lahendab ülesande lõpuni. Programm kontrollib lahendusefaili olemasolu ja vajadusel loob uue faili (fail luuakse samasse kausta, kus on lahendusprogramm). Programm seab lahendusfaili nimeks <ülesandenimi_><lahendaja nimi>.Lahendaja nimeks on see nimi, mille kasutaja sisestab programmi avades. Kõik samanimelise ülesande lahenduskäigud salvestatakse ühte faili (ainult sellisel juhul, kui kasutajanimi on sama).

Näiteks:

```
=UUS Joosep 2013/05/07 14:13:53
jagamine_murd
Mitu õpilast õpib teises klassis?
(a)+(b)
5.0 + 9.0 14.0
Mitu õpilast õpib kolmandas klassis?
(c)+(d)
9.0 + 10.0 19.0
Mitu õpilast õpib teises ja kolmandas klassis?
(a+b)+(c+d)
14.0 + 19.0 33.0
Mitu mandariini sai iga õpilane?
(e):(a+b+c+d)
200.0 : 33.0 6.1
Suurepärase! Ülesanne lahendatud. . 14:16:20
```

2.3. Vigade salvestamine

Lahendamise käigus on kasutajal võimalik teha nelja tüüpi vigu:

- Küsimuse koostamise vead:
 - Kui küsimusele on vastus juba olemas.

- Kui küsimus ei ole mõistlik – ehk sellisele küsimusele ei ole vastavusse pandud ühtegi avaldist.
- Kui küsimusele ei ole võimalik ühe sammuga vastust leida.
- Avaldise koostamise vigu:
 - Avaldisest puudub tehtemärk.
 - Avaldisest puudub esimene operand.
 - Avaldisest puudub teine operand.
 - Moodustatud avaldis ei vasta küsimusele.
- Arvutamise vigu
 - Kui vastus on valel kujul (näiteks kasutaja on sisestanud mitu „“ vastusesse).
 - Kasutaja sisestatud vastus on vale.
- Vastuse andmise vigu
 - Kasutaja soovib lõppvastust anda enne, kui see on leitud.
 - Kasutaja soovib ülesannet edasi lahendada pärast lõppvastuse leidmist.

Rakendus loob kasutaja vigade salvestamiseks eraldi spetsiifilise kujuga vigadefaili. Vigadefaili salvestatakse iga kasutaja lahenduse käigus tehtud vead (Kasutaja peab ülesande edukalt lahendama.). Vigadefail on kõigil kasutajatel ühine (eeldusel, et rakendus käivitati samast kaustast).

Näiteks:

Ül	küsimuse koostamise	arvutamise	avaldise koostamise	vastuse andmise vigu
joosepk: Näidisülesanne	0	0	0	0
test2d: Näidisülesanne	1	2	0	3
abctest: Näidisülesanne	0	0	0	0
test k: Näidisülesanne	0	0	0	0
joosepk: Näidisülesanne	4	4	2	1
Joosepk: Näidisülesanne	3	0	0	0
joosepk: Näidisülesanne	0	2	0	2
joosepK: jagamine_murd	2	3	3	1

Vead on faili kirjutatud tabeli kujul. Esimesel real on kirjas veatüübid. Kasutaja vead kirjutatakse vigadefaili ainult siis, kui kasutaja lahendab ülesande edukalt (leiab lõppvastuse). Programm kontrollib vigadefaili olemasolu ja vajaduse korral loob uue

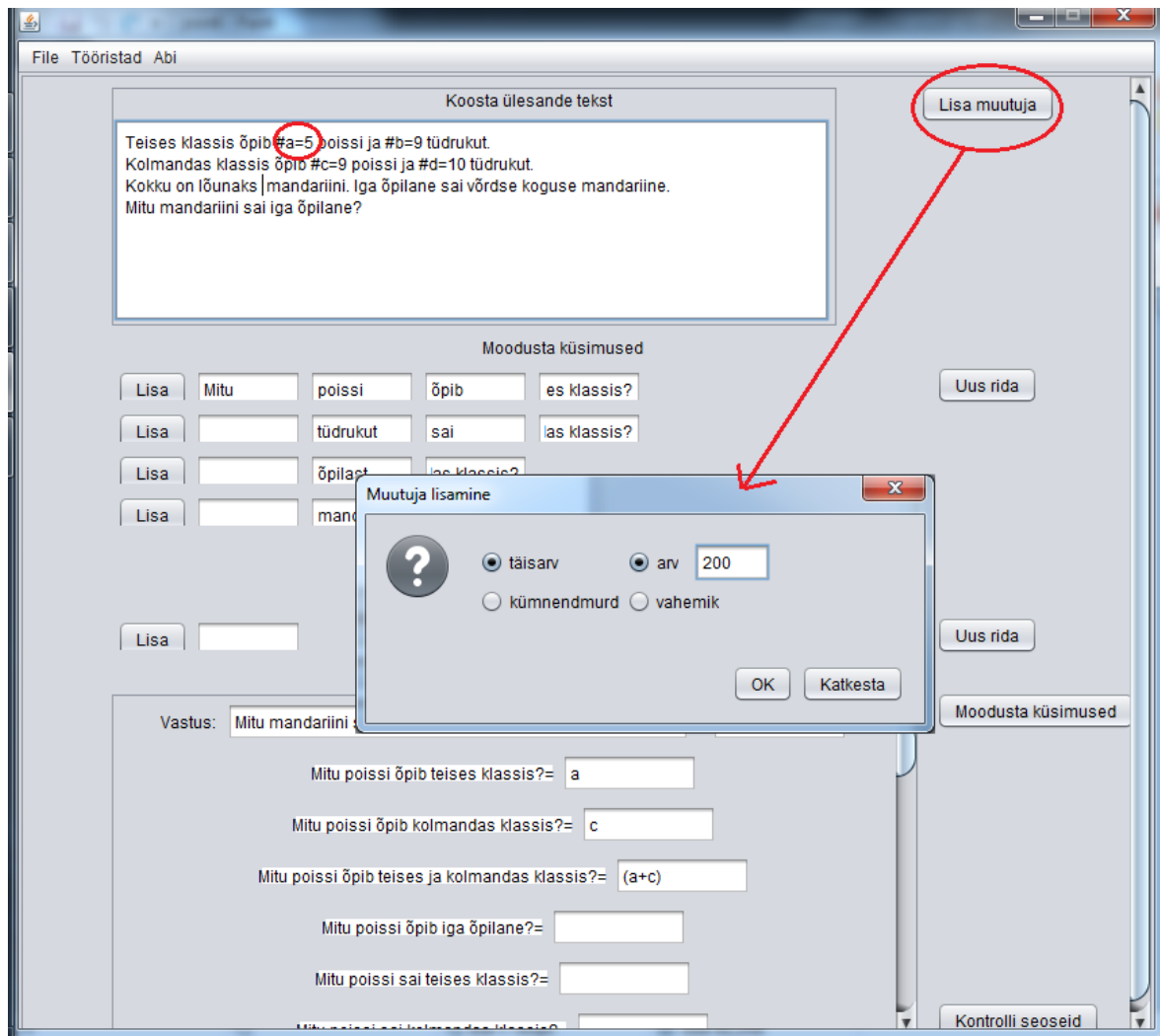
vigadefaili (fail luuakse lahendusprogrammiga samasse kausta). Lisaks kirjutab programm kasutaja vead ka kasutaja lahendusfaili.

2.4. Tekstülesannete koostamise liides

Koostamise programm on disainitud nii, et ülesannet saaks koostada võimalikult veavabalt. E.Koppeli poolt loodud programm ei sisaldanud liidest ülesannete koostamiseks (õpetaja poolt). Õpetaja pidi mingi tekstiredaktori abil ülesandeteksti spetsiifilisel kujul kirjutama tekstifaili ja väiksemgi viga tähendas seda, et lahendamise programm luges ülesande vigaseks (näiteks õpetaja tegi seoste kirjutamisel mingi lohakusevea) või mõistis ülesannet valesti.

2.4.1. Ülesande teksti koostamine ja muutujate lisamine

Ülesande koostaja kirjutab ülesande akna ülemises osas asuvale tekstialale (joonist 8). Kasutaja saab ülesande teksti muutujaid lisada vajutades „Lisa muutuja“ nupu peale (joonist 8). Avanevas paneelis on võimalik kasutajal valida, kas lisatav muutuja on täisarv või kümnendmurd. Lisaks on võimalik valida, kas arv on konstant või genereeritakse arv mingis vahemikus. Täisarv vahemikus ja kümnendmurd vahemikus genereeritakse juhuslikult kasutaja poolt etteantud vahemikus.



Joonis 8. Muutuja lisamine tekstialasse

Muutujad on arvulised suurused, mis kirjutatakse ülesande teksti kujul „#<muutuja tähistus>=<muutuja väärtus>“, kus igale suurusele vastab unikaalne muutuja (joonis 8). Muutuja tähistus on väike ladina täht. Komakoha tähistajana kasutab programm kümnendpunkti. Programm ei võimalda sisestada vahemikku ,mille algus on suurem kui lõpp. Rakendus väljastab kasutajale hoiatuse, kui kasutaja sisestab muutuja väärtuseks nulli. Kasutaja saab sisestada nupuga „Lisa muutuja“ kuni 26 muutujat (a-z). Juhul kui üritatakse sisestada rohkem muutujaid väljastab rakendus veateate. Kasutaja saab tekstialal olevat muutuja väärtust muuta, kui asendab võrdusmärgi taga oleva numbri mingi muu numbriga (joonis 8). Kasutaja saab muutuja eemaldada kustutades selle tekstialalt.

2.4.2. Ülesande küsimuste moodustamine

Küsimuste moodustamiseks on programmis kaks paneeli, mis võimaldavad korraga erineva pikkuse ja sisuga küsimuste moodustamist. Kasutaja kirjutab paneelil olevatesse

tekstiväljadesse küsimuse osasid (joonis 9), mille abil lahendamise programmis õpilane saab moodustada küsimusi. Küsimused on võimalik valmis genereerida, vajutades nupule „Moodusta küsimused“.

File Tööriistad Abi

Teises klassis õpib #a=5 poissi ja #b=9 tüdrukut.
Kolmandas klassis õpib #c=9 poissi ja #d=10 tüdrukut.
Iga õpilane saab lõunaks #e=2 mandariini.
Mitu mandariini said õpilased kokku?

Moodusta küsimused

Lisa Mitu poissi õpib es klassis? Uus rida

Lisa tüdrukut las klassis?

Lisa õpilast las klassis?

Lisa

Lisa 1 mandariini said poisid es klassis? Uus rida

Lisa Mitu mandariini las klassis?

Lisa said õpilased las klassis?

Lisa

Lisa

Lisa

Sea (mõistlikele) küsimustele vastavusse avaldised

Vastus: Mitu mandariini said õpilased teises ja kolmandas klassis? = $(a+b+c+d)*e$ Moodusta küsimused

Mitu poissi õpib teises klassis? = a

Mitu poissi õpib kolmandas klassis? = c

Joonis 9. Küsimuste moodustamine koostamise programmis

Ühe küsimuste moodustamise paneeliga saab teha kuni kuuest elemendist koosnevaid küsimusi. Iga küsimuse elementide grupi suuruseks on kuni kuus elementi (joonis 9). Paneeli iga veerg on üks elementide grupp. Programm ei lase lisada üle kuue veeru ja igas reas on maksimaalselt kuus tekstivälja. Kui kasutaja on reale lisanud maksimaalse arvu tekstivälju või lisanud maksimaalse arvu ridu, siis muudab programm vastava(d) nupu(d) mitteaktiivseks (joonis 9). Vajutades nupule „Moodusta küsimused“ genereerib programm küsimuste elementide põhjal kõikvõimalikud maksimaalse pikkusega (joonis 7) küsimused valmis ja lisab seoste moodustamise paneelile. Programm leiab maksimaalse küsimuse pikkuse loendades moodustamise paneelilt kasutaja poolt sisestatud elementide veergude arvu, näiteks kui kasutaja on sisestanud elemente nelja veergu, on maksimaalne pikkus neli elementi.

2.4.3. Küsimustele vastavate avaldiste ja vastuse moodustamine

Ülesande koostaja moodustab lõppvastuse ja sellele vastava avaldise ning lisab mõistlikele programmi poolt genereeritud küsimustele avaldised.

File Tööriistad Abi

Koosta ülesande tekst

Teises klassis õpib #a=5 poissi ja #b=9 tüdrukut.
Kolmandas klassis õpib #c=9 poissi ja #d=10 tüdrukut.
Kokku on lõunaks #e=200 mandariini. Iga õpilane sai võrdse koguse mandariine.
Mitu mandariini sai iga õpilane?

Lisa muutuja

Moodusta küsimused

Lisa Mitu poissi õpib es klassis?

Lisa tüdrukut sai las klassis?

Lisa õpilast las klassis?

Lisa mandariini ga õpilane?

Uus rida

Uus rida

Sea (mõistlikele) küsimustele vastavusse avaldised

Vastus: Mitu mandariini sai iga õpilane? = e:(a+b+c+d)

Mitu poissi õpib teises klassis?= a

Mitu poissi õpib kolmandas klassis?= c

Mitu poissi õpib teises ja kolmandas klassis?= a+c

Mitu poissi õpib iga õpilane?=

Mitu poissi sai teises klassis?=

Mitu poissi sai kolmandas klassis?=

Moodusta küsimused

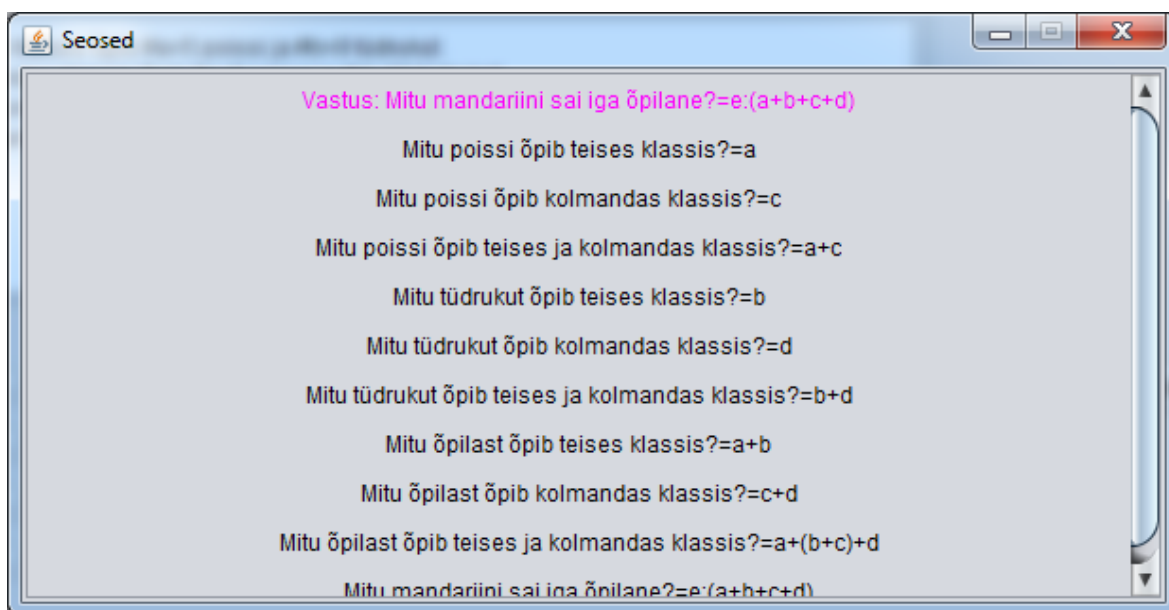
Kontrolli seoseid

Joonis 6. Moodustatud avaldiste kontrollimine

Vastuse väli on tähistatud tekstiga „Vastus:“. Võrdusmärgist vasakule poole peab kasutaja sisestama lõppvastusele vastava küsimuse ja paremale poole lõppvastusele vastava avaldise (joonis 8). Programm kontrollib küsimuse ja vastuse olemasolu ning juhul, kui üks või mõlemad neist puuduvad, väljastab salvestamisel veateate. Kasutaja saab genereeritud küsimustele vastava avaldise kirjutada vastava küsimuse taha asuvale tekstiväljale. Kasutaja võib alati uuesti moodustada küsimused vajutades nupule „Moodusta küsimused“, mille tagajärjel programm eemaldab kasutaja poolt kirjutatud avaldised ja vastuse. Avaldised tuleb moodustada kasutades programmi poolt genereeritud muutujate tähistusi (joonis 8, joonis 11). Programm väljastab salvestamisel veateate, kui kasutaja pole sisestanud ühelegi küsimusele vastavat avaldist.

2.4.3 Moodustatud seoste kuvamine

Kasutajal on võimalik eraldi paneelilt näha tema poolt moodustatud seoseid. Seoste vaatamiseks peab kasutaja vajutama nupule „Kontrolli seoseid“. Rakendus avab uues aknas seostepaneeli, milles on näha kõik kasutaja poolt moodustatud seosed (joonis 12). Lõppvastust tähistav seos on esimene ja teistest erinevat värvi, et võimaldada vastuse lihtsamat eristamist muudest küsimustest.



Joonis 7. Moodustatud seoste kuvamine

Seoste paneelil näidatakse ainult need seosed (küsimus ja temale vastav avaldis), mille kasutaja avaldiste moodustamise juures mingid avaldised moodustas. Rakendus kontrollib, et avaldise oleks õigelt kuju (Avaldisse lisatud muutujad peavad olema olema, peab kasutama „+“, „-“, „*“, „/“ tehtemärke ja sulge peab olema võrdselt).

2.4.4. Ülesande salvestamine

Kasutaja saab ülesannet salvestada vajutades menüüs „File“ valikule „Salvesta“. Rakendus avab salvestamise akna (joonis 13) ja võimaldab kasutajal valida ülesande salvestamise kausta. Reale „File Name“ tuleb kasutajal sisestada ülesande nimi. Rakendus salvestab ülesande automaatselt faililaiendiga „.txt“ (ka juhul kui kasutaja sisestab mingi muu faililaiendi või jätab faililaiendi sisestamata).

2.4.5. Ülesande avamine

Kasutaja saab koostamise programmiga olemasolevat ülesandefaili avada vajutades menüüs „File“ valikule „Ava“. Rakendus avab faili otsimise akna (sarnane salvestamise aknaga). Kasutajal on võimalik valida millist ülesandefaili ta soovib avada. Ülesandefail peab olema faililaiendiga „txt“ ja kindla kujundusega tekstifail. Programm väljastab veateate, kui avatav fail ei ole õige kujundusega. Rakendus loeb avatavast ülesandest vajalikud andmed sisse ja kirjutab need vastavatesse väljadesse. Ülesande lõppvastus loetakse sisse ainult sellisel juhul, kui lõppvastuse küsimusele vastav avaldis on võrdne ülesandes antud vastuse avaldisega.

2.4.6. Uue ülesande loomine

Koostamise programmi käima pannes kuvatakse kasutajale uue ülesande loomise vaade. Lisaks on kasutajal võimalik uut ülesannet alustada vajutades menüüs „File“ valikule „Uus“ (joonis 10). Rakendus eemaldab poolelioleva ülesande ja kuvab kasutajale uue ülesande loomise vaate.

2.4.7 Muutujate genereerimine

Kui ülesande koostaja on ülesande moodustamise ajal muutuja loomise ajal valinud muutuja väärtuste jaoks vahemiku, siis lahendusprogramm selle ülesande avamisel genereerib vastava muutuja arvulised väärtused kasutaja määratud vahemikust.

3. Ülesande esitus tekstifailis

Iga ülesanne salvestatakse spetsiaalse kujuga tekstifaili. Tekstifaili laiendiks on .txt. Ülesandefaili ülesehituses üritati võimalikult palju kasutada E. Koppeli poolt loodud formaati [1], tehes vajadusel sinna muudatusi.

Failis peab olema:

1. Ülesande tekst ja muutujad.
2. Küsimuste moodustamise maatriksi veerud.
3. Muutujatest moodustatud avaldised ja neile vastavad küsimused.
4. Vastuse avaldis.

Lisaks võivad veel failis olla:

- Muutujate vahemiku informatsioon
- Küsimuste moodustamise lisamaatriksi veerud

Näiteks:

--TEKSTER2--

!SEADED=

#a:5:12:0

#b:8:15:0

#c:6:12:0

#d:9:20:0

!TEKST=

Teises klassis õpib #a=8 poissi ja #b=9 tüdrukut.

Kolmandas klassis õpib #c=7 poissi ja #d=10 tüdrukut.

Mitu õpilast õpib teises ja kolmandas klassis kokku?

Iga õpilane saab lõunaks #e=2 mandariini.

Mitu mandariini said õpilased kokku?

!VEERUD=

Mitu

poissi&tüdrukut&õpilast

õpib

teises klassis?&kolmandas klassis?&teises ja kolmandas klassis?

!VEERUD2=

Mitu mandariini

said poisid&said tüdrukud&said õpilased

teises klassis?&kolmandas klassis?&teises ja kolmandas klassis?

!VASTAVUSED=

a=Mitu poissi õpib teises klassis?

b=Mitu tüdrukut õpib teises klassis?

c =Mitu poissi õpib kolmandas klassis?
 d =Mitu tüdrukut õpib kolmandas klassis?
 $(a+c)$ =Mitu poissi õpib teises ja kolmandas klassis?
 $b+d$ =Mitu tüdrukut õpib teises ja kolmandas klassis?
 $a+b$ =Mitu õpilast õpib teises klassis?
 $c+d$ =Mitu õpilast õpib kolmandas klassis?
 $a+(b+c)+d$ =Mitu õpilast õpib teises ja kolmandas klassis?
 $(a+(b+c)+d)*e$ =Mitu mandariini said õpilased teises ja kolmandas klassis?
 $(a+b)*e$ =Mitu mandariini said õpilased teises klassis?
 $(c+d)*e$ =Mitu mandariini said õpilased kolmandas klassis?
 $a*e$ =Mitu mandariini said poisid teises klassis?
 $c*e$ =Mitu mandariini said poisid kolmandas klassis?
 $(a+c)*e$ =Mitu mandariini said poisid teises ja kolmandas klassis?
 $b*e$ =Mitu mandariini said tüdrukud teises klassis?
 $d*e$ =Mitu mandariini said tüdrukud kolmandas klassis?
 $(b+d)*e$ =Mitu mandariini said tüdrukud teises ja kolmandas klassis?
 !VASTUS=
 $(a+(b+c)+d)*e$

Ülesande komponendid peavad algama ülesandefailis järgmiste ridadega:

1. Ülesanne algab reaga „-=TEKSTER=-„.
2. Muutujate kohta käiv osa algab reaga „!SEADED=“. Järgmistel ridadel on igal real mingi muutuja kohta käiv informatsioon kujul <muutuja tähistus>:<vahemiku algus>:<vahemiku lõpp>:<arvu või kümnendmurrutähis (on „0“ kui tegu on täisarvuga ja „1“ kui tegu on kümnendmurruga)>. Muutujate kohta käiv osa võib ülesandefailist puududa.
3. Ülesande teksti osa algab järgmise reaga „!TEKST=“
4. Küsimuste moodustamise maatriksi osa algab reaga „!VEERUD=“.
5. Küsimuste moodustamise lisamaatriks algab reaga „!VEERUD2=“.
6. Seoste osa algab reaga „!VASTAVUSED“
7. Lõppvastuse osa algab reaga „!VASTUS=“.

Ülesande koostamise programm kontrollib ainult kas ülesanne algab reaga „-=TEKSTER2=-„. Eespool nimetatud tingimuste mitte täitmise korral väljastab rakendus vastava veateate. Andmete sisselugemisel kontrollib koostamise rakendus vastavate osade olemasolu, aga ei väljasta puuduvate osade korral veateateid (välja arvatud esimese tingimuse korral). Lõpetamata ülesandefailid on võimalik uuesti avada ja jätkata ülesande koostamisega

Programmi ja andmestruktuuride lühike kirjeldus

Programmid on kirjutatud programmeerimiskeeles Java.

Programmide käivitamiseks on vajalik JVM (Java Virtual Machine) olemasolu kasutatavas arvutis. JVM on tasuta kättesaadav Oracle kodulehelt [4].

Lahendamise programm koosneb 10 java failist ja ühest .ini failist. Java failid on kompileeritud ning kokku pandud „tekster.jar“ nimeliseks rakenduseks.

Koostamise programm koosneb kolmest java failist. Java failid on kompileeritud ning kokku pandud „teksterTeacher.jar“ nimeliseks rakenduseks.

Lahendamise programmis lisati

1. WrapLayout [5] klass ülesandeteksti korralikumaks kuvamiseks.

Õpetajaprogrammi graafilise külje loomise juures on kasutatud ka integreeritud programmeerimiskeskonna (*IDE Integrated Development Environment*) võimalusi [6-8].

Koostamise programmis on järgmised failid:

1. MainClass – õpetajaprogrammi peaklass, kus toimub rakenduse tööle panek.
2. SolutionConstructor – õpetajaprogrammi klass, kus pannakse kokku rakenduse graafiline külg ja funktsionaalsus
3. JfileFilter – ülesandefailide avamiseks ja salvestamiseks loodud filter, määrab ülesandefaili formaadi.
4. Utils – ülesandefaili salvestamisel faililaiendi kontrollimise klass
5. ExpressionParser [10] – klass, milles seoste kontrollimise ajal avaldised viiakse pööratud poola kujule ja arvutatakse nende vastus.

Ülesande teksti, muutujaid, tekstist tulenevaid küsimusi ja neile vastavaid avaldisi, küsimuste moodustamise maatriksit (või maatrikseid), lõppvastust ja muutujate kohta käivat informatsiooni hoitakse tekstifailides. Samuti hoitakse tekstifailides kasutaja lahenduskäiku ja kasutaja poolt tehtud vigade loetelu.

4. Programmi edasi arendamise võimalused

Lahendamise programmile on lisaks võimalus lisada:

1. Pikema avaldisega sammud – lahendamise rakendusega saab korraga teha samme mis eeldavad rohkem kui ühe tehte sooritamist korraga.
2. Konstandid – osades ülesannetes tuleb mingeid suurusid korrutada näiteks kolmega või teisendada ühikuid, samuti on füüsika ja keemia ülesannete lahendamisel paljud suurused fikseeritud konstandid, mida iga ülesande tekstis ei esitata.
3. Automaatlahendus – kasutajale näidatakse ette lahenduskäik.
4. Sammunõuanded – kasutajale soovitatakse, millist sammu järgmisena teha.
5. Õpilase lahenduse hinnang – kasutaja lahenduskäigu võrdlemine optimaalse lahenduskäiguga.
6. Üleliigsete sammude kustutamine – kui kasutaja lahenduskäik sisaldab samme, mida kasutaja lahenduseni jõudmiseks hiljem ei kasuta, on võimalik need sammud kustutada.
7. Negatiivsete arvudega arvutamine – olemasolevas rakenduses on võimalik arvutada ainult positiivsete arvudega.
8. Vigadefaili ja lahendustefaili krüptimine – takistada õpilastel pärast ülesande lahendamist oma lahenduskäigu muutmist.

Koostamise programmile on lisaks võimalik lisada:

1. Lahendaja lahenduskäigu kontrollimine- lahendusfaili on võimalik kontrollida, et tuvastada lahendusjärgsed muudatused.
2. Vigade ja lahenduskäigu kuvamine – koostamise rakendusega on võimalik avada vigadefaili ja lahenduskäiku ja neid vaadata.
3. Vigade analüüsimine – võimalik koostada vigade põhjal statistikat.

Adding functionality to word problem solving environment.

Bachelor thesis
Joosep Kibal

Abstract

The result of this bachelor work is an improved word problem solving environment. The basic version of the environment was made by E. Koppel in 2006. The added functionality includes:

1. Adding division arithmetic operation to the environment.
2. Adding decimal fraction calculation capability.
3. Adding the capability to save user's solution and mistakes made during the solving.
4. Adding the functionality to generate variables used in exercises.
5. Creation of teacher's program for creating word problems.

The environment for solving word problems is aimed for elementary school children to help them solve word problems and familiarise them with using the computer. Current environment allows the user to solve word problems by calculating meaningful quantities step by step and then reaching the answer. The current constraints of the environment are that only mathematical operations with positive numbers are allowed.

Viited

Kõik viited on kontrollitud 11.05.13 seisuga.

1. E.Koppel. Tekstülesannete lahendamise programmi baasvariant . Bakalaureusetöö. 2006.
2. Arvud ja algebra kümnendmurrud;
<http://www.miksike.ee/documents/main/lisa/probleem/m1kumnendmurrud.htm>
3. Merit software Word Problem Shape-Up;
http://www.meritsoftware.com/software/word_problem_shape_up/
4. Java™Platform, Standard Edition 7 API Specification;
<http://docs.oracle.com/javase/7/docs/api/index.html>
5. Java Tips Webblog, Wrap Layout;
<http://tips4java.wordpress.com/2008/11/06/wrap-layout/>
6. Create Great-Looking GUIs With NetBeans IDE;
<http://www.oracle.com/technetwork/articles/javase/nb-guibuilder-136926.html>,
7. Learning Swing with the NetBeans IDE;
<http://docs.oracle.com/javase/tutorial/uiswing/learn/>,
8. Introduction to GUI Building;
<https://netbeans.org/kb/docs/java/gui-functionality.html>
9. Ülesandekogu;
<http://www.miksike.ee/documents/main/lisa/5klass/6raamat/matemaatika/5-6-4-3m.htm>
10. Converting infix to RPN (shunting-yard algorithm);
<http://andreinc.net/2010/10/05/converting-infix-to-rpn-shunting-yard-algorithm/>

Lisa 1 (CD)

1. Programmi kood
2. Ülesannetekogu

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Joosep Kibal
(sünnikuupäev: 16.10.1990)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose , “Tekstülesannete lahendamise programmi funktsionaalsuse täiendamine“ , mille juhendaja on Rein Prank
 - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus , **13.05.2013**